

**In the Claims:**

Please amend claims 1, 4, 11, 12, 17, 18, 19, and 20 as follows:

1. (currently amended) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array and a plurality of calendars for scheduling said flows, said scheduling method comprising the steps of:

storing an active flow indicator for each calendar entry in a calendar status array (CSA);

storing a subset of said active flow indicators from said calendar status array (CSA) in a cache utilizing a current pointer (CP) to a calendar entry and loading said subset of said active flow indicators from said calendar status array (CSA) starting at said current pointer (CP) CSA entry; and

updating said calendar status array (CSA) based upon a predefined calendar range and resolution; and

utilizing said subset of said active flow indicators stored in said cache to determine a flow of a calendar for servicing.

2. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 1 wherein the step of storing said active flow indicator for each calendar entry in said calendar status array (CSA) includes the step of storing one bit for each calendar entry in said calendar status array (CSA).

3. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 1 wherein the step of storing said active flow indicator for each calendar entry in said calendar status array (CSA) includes the step of storing said active flow indicator for each weighted fair queue (WFQ) ring entry in a calendar status array (CSA).

4. (currently amended) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 1 wherein said cache includes a predefined number of bits for storing said cache copy subset for each of the calendars and wherein the step of storing said subset of said active flow indicators from said calendar status array (CSA) in a cache utilizing a current pointer (CP) to a calendar entry and loading said subset of said active flow indicators from said calendar status array (CSA) starting at said current pointer (CP) CSA entry and includes the step of utilizing a current pointer (CP) to a calendar entry and loading said subset of said active flow indicators from said calendar status array (CSA) starting at said current pointer (CP) CSA entry and increasing through said predefined number of CSA entries equal to said predefined number of bits.

5. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 4 wherein the step of loading said subset of said active flow indicators from said calendar status array (CSA) includes the step of using said current pointer (CP) to determine one of plurality of addressed portions of said calendar status array (CSA) to be accessed.

6. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 5 wherein said predefined number of bits for storing said subset for each of the calendars is thirty-two bits and wherein said calendar status array (CSA) includes four addressed portions of one hundred twenty-eight bits of said calendar status array (CSA), each said addressed portion accessible in one cycle.

7. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 1 wherein the step of storing said active flow indicator for each calendar entry in said calendar status array (CSA) includes the step for each of said plurality of calendars of storing said active flow indicator for each calendar entry in said calendar status array (CSA).

8. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 7 wherein said plurality of calendars include a low latency service (LLS) calendar, a normal latency service (NLS) calendar, and a peak bandwidth service (PBS) calendar and includes the step segmenting each of said plurality of calendars into epochs (0:p), where epoch 0 has a highest resolution and a lowest range; and epoch 1 through epoch p have a range of n(p) times a range of epoch 0; and epoch 1 through epoch p have a resolution of  $1/n^{**}p$  times a resolution of epoch 0, where n is a scaling factor.

9. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 8

wherein the step of updating said calendar status array (CSA) based upon a predefined calendar range and resolution includes the step of updating said calendar status array (CSA) based upon said predefined calendar range and resolution of said epoch 0 through said epoch p.

10. (original) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 8 wherein the step of updating said calendar status array (CSA) based upon said predefined calendar range and resolution of said epoch 0 through said epoch p for said low latency service (LLS) calendar, said normal latency service (NLS) calendar, and said peak bandwidth service (PBS) calendar.

11. (currently amended) A scheduling method for implementing Quality-of-Service (QoS) scheduling for a plurality of flows with a cached status array as recited in claim 1 includes the step of utilizing said subset of said active flow indicators for incrementing a said current pointer (CP) by an identified number of positions up to a current time (CT) value, said identified number of positions equal to a variable number of inactive flow indicators up to said current time (CT) value and said identified number of positions having a maximum value equal to a number of entries in said cache.

12. (currently amended) A QoS scheduler for implementing Quality-of-Service (QoS) scheduling for a plurality of flows comprising:

a queue manager;

a plurality of calendars coupled to said queue manager for scheduling said flows, a calendar status array (CSA) coupled to said plurality of calendars for storing an

active flow indicator for each calendar entry for each of said plurality of calendars;  
a cache coupled to calendar status array (CSA) for storing a cache copy subset  
of said active flow indicators from said calendar status array (CSA) utilizing a current  
pointer (CP) to a calendar entry and for loading said subset of said active flow  
indicators from said calendar status array (CSA) starting at said current pointer (CP)  
CSA entry; and

a calendar updating algorithm for updating said calendar status array (CSA)  
based upon a predefined calendar range and resolution; and  
said queue manager for utilizing said cache copy subset of said active flow  
indicators to determine a flow of a calendar for servicing.

13. (original) A QoS scheduler for implementing Quality-of-Service (QoS)  
scheduling for a plurality of flows as recited in claim 12 wherein plurality of calendars  
include a low latency service (LLS) calendar, a normal latency service (NLS) calendar,  
and a peak bandwidth service (PBS) calendar.

14. (original) A QoS scheduler for implementing Quality-of-Service (QoS)  
scheduling for a plurality of flows as recited in claim 12 wherein said calendar updating  
algorithm for updating calendar status array (CSA) based upon a predefined calendar  
range and resolution includes segmenting each of said plurality of calendars into  
epochs (0:p), where epoch 0 has a highest resolution and a lowest range; and epoch 1  
through epoch p have a range of n(p) times a range of epoch 0; and epoch 1 through  
epoch p have a resolution of  $1/n^{**}p$  times a resolution of epoch 0, where n is a scaling  
factor.

15. (original) A QoS scheduler for implementing Quality-of-Service (QoS) scheduling for a plurality of flows as recited in claim 14 wherein said calendar updating algorithm for updating calendar status array (CSA) based upon a predefined calendar range and resolution is based upon a priority of said low latency service (LLS) calendar, said normal latency service (NLS) calendar, and said peak bandwidth service (PBS) calendar and a number of said epoch.

16. (original) A QoS scheduler for implementing Quality-of-Service (QoS) scheduling for a plurality of flows as recited in claim 14 wherein said calendar updating algorithm for updating calendar status array (CSA) based upon a predefined calendar range and resolution accesses said calendar status array (CSA) by an ascending order said epoch for each of said calendars in an order of said low latency service (LLS) calendar, said normal latency service (NLS) calendar, and said peak bandwidth service (PBS) calendar.

17. (currently amended) A QoS scheduler for implementing Quality-of-Service (QoS) scheduling for a plurality of flows as recited in claim 14 includes a said current pointer (CP) to determine one of a plurality of addressed portions of said calendar status array (CSA) to be accessed in a system cycle.

18. (currently amended) A QoS scheduler for implementing Quality-of-Service (QoS) scheduling for a plurality of flows as recited in claim 14 includes current pointer (CP) incrementing means utilizing said cache copy subset of said active flow indicators for incrementing a said current pointer (CP) by an identified number of positions up to a current time (CT) value, said identified number of positions equal to a variable number

of inactive flow indicators up to said current time (CT) value and said identified number of positions having a maximum value equal to a number of entries in said cache.

19. (currently amended) A computer program product for implementing Quality-of-Service (QoS) scheduling of a plurality of flows with a cached status array and a plurality of calendars for scheduling said flows in a scheduler, said computer program product including a plurality of computer executable instructions stored on a computer readable medium, wherein said instructions, when executed by said scheduler, cause said scheduler to perform the steps of:

storing an active flow indicator for each calendar entry for each of said plurality of calendars in a calendar status array (CSA);

storing a subset of said active flow indicators from said calendar status array (CSA) for each of said plurality of calendars in a cache utilizing a current pointer (CP) to a calendar entry and loading said subset of said active flow indicators from said calendar status array (CSA) starting at said current pointer (CP) CSA entry; and

updating said calendar status array (CSA) based upon a predefined calendar range and resolution for each of said plurality of calendars; and

utilizing said subset of said active flow indicators for each of said plurality of calendars to determine a flow of a calendar for servicing.

20. (currently amended) A computer program product for implementing Quality-of-Service (QoS) scheduling of a plurality of flows with a cached status array and a plurality of calendars for scheduling said flows in a scheduler as recited in claim 19 wherein said instructions, when executed by said scheduler, cause said scheduler to

perform the steps of utilizing said subset of said active flow indicators for incrementing a said current pointer (CP) by an identified number of positions up to a current time (CT) value, said identified number of positions equal to a variable number of inactive flow indicators up to said current time (CT) value and said identified number of positions having a maximum value equal to a number of entries in said cache.